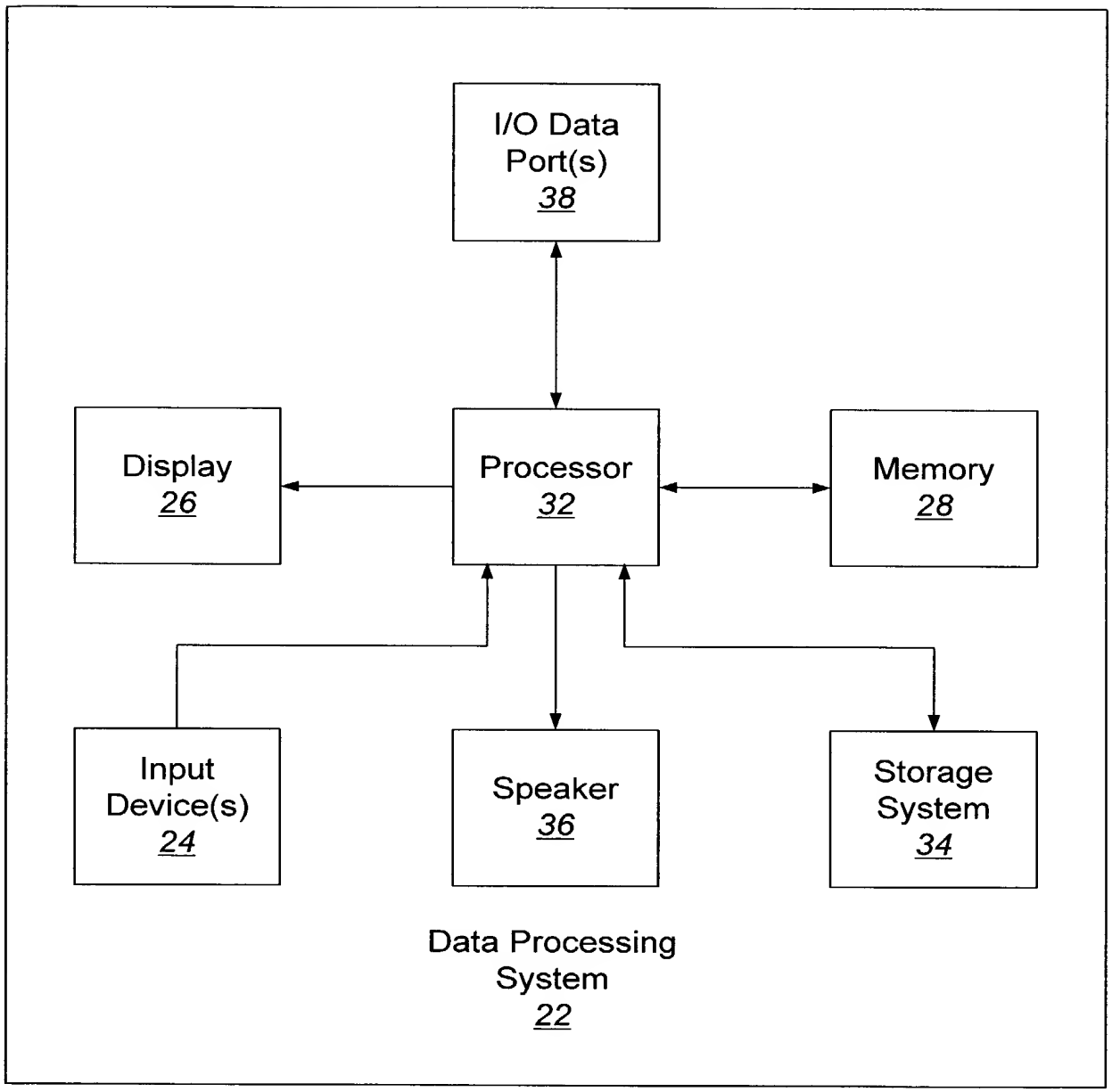

I/O Data Port(s) 38
Display 26
Processor 32
Memory 28
Input Device(s) 24
Speaker 36
Storage System 34
Data Processing System 22

FIG. 1

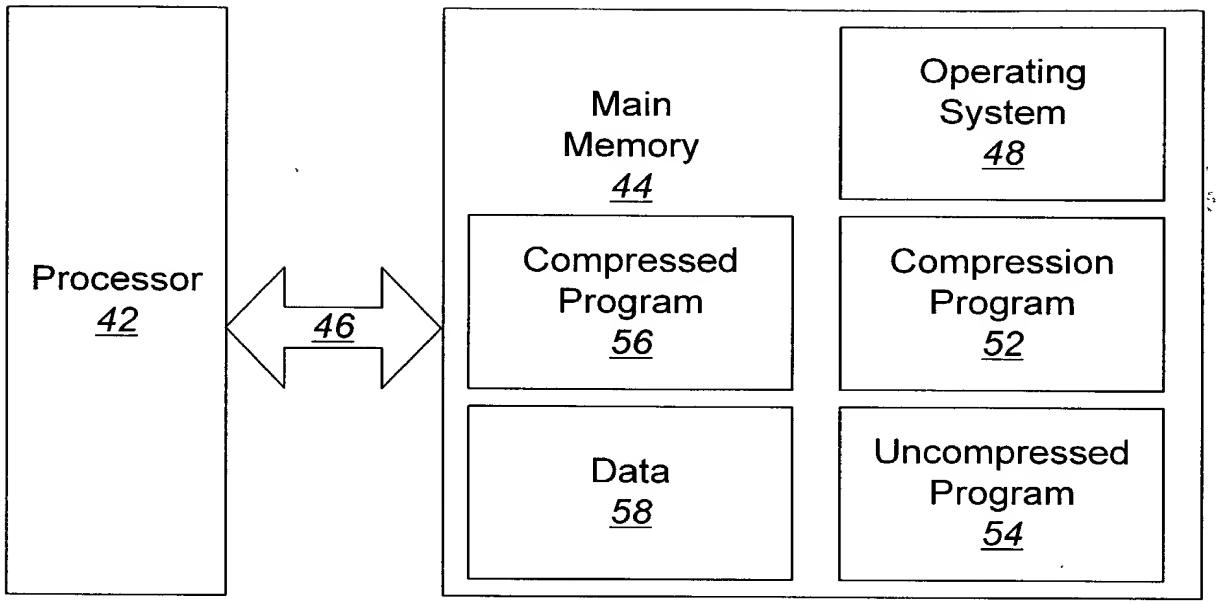


FIG. 2

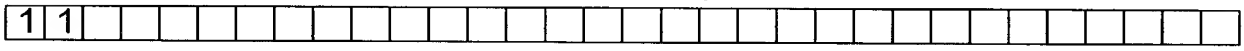
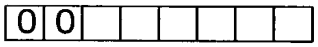


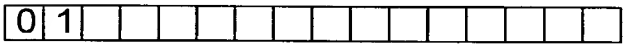
FIG. 3A

Uncompressed Instruction



Type I Compressed Instruction

FIG. 3B



Type II Compressed Instruction

FIG. 3C

Type III Compressed Instruction

FIG. 3D

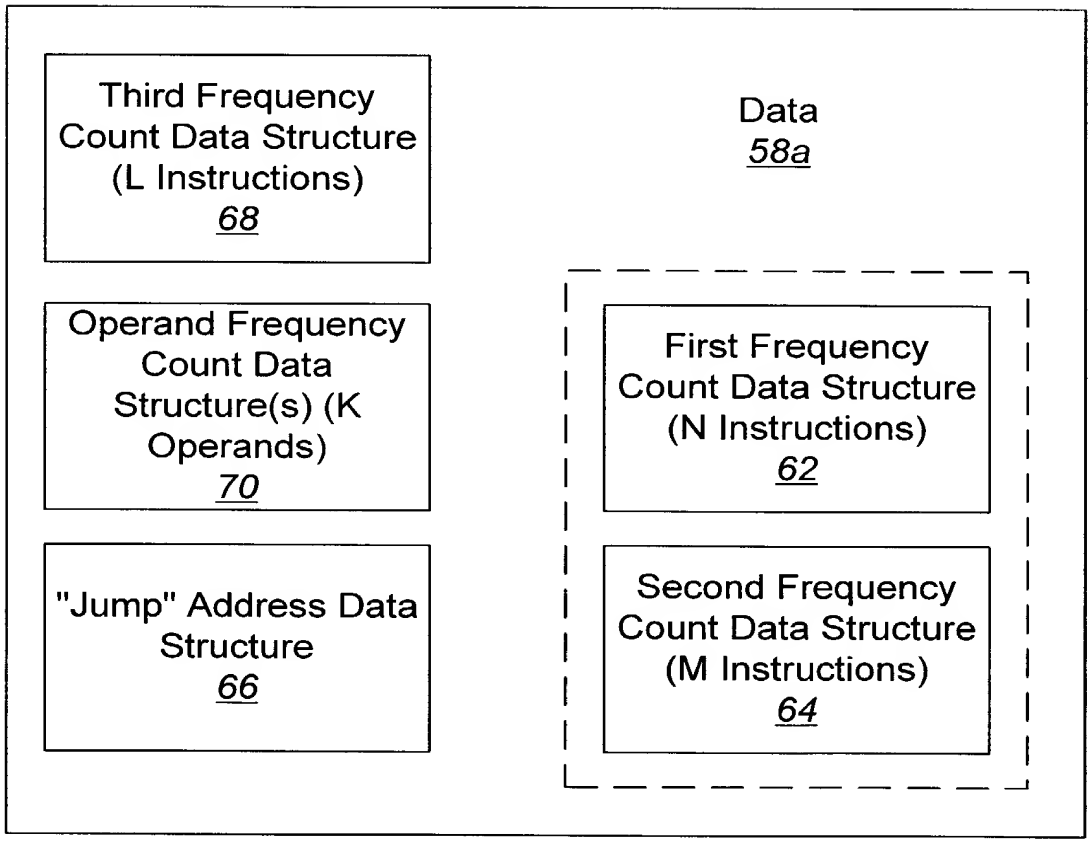

Third Frequency Count Data Structure (L Instructions) 68
Data 58a
Operand Frequency Count Data Structure(s) (K Operands) 70
First Frequency Count Data Structure (N Instructions) 62
"Jump" Address Data Structure 66
Second Frequency Count Data Structure (M Instructions) 64

FIG. 4

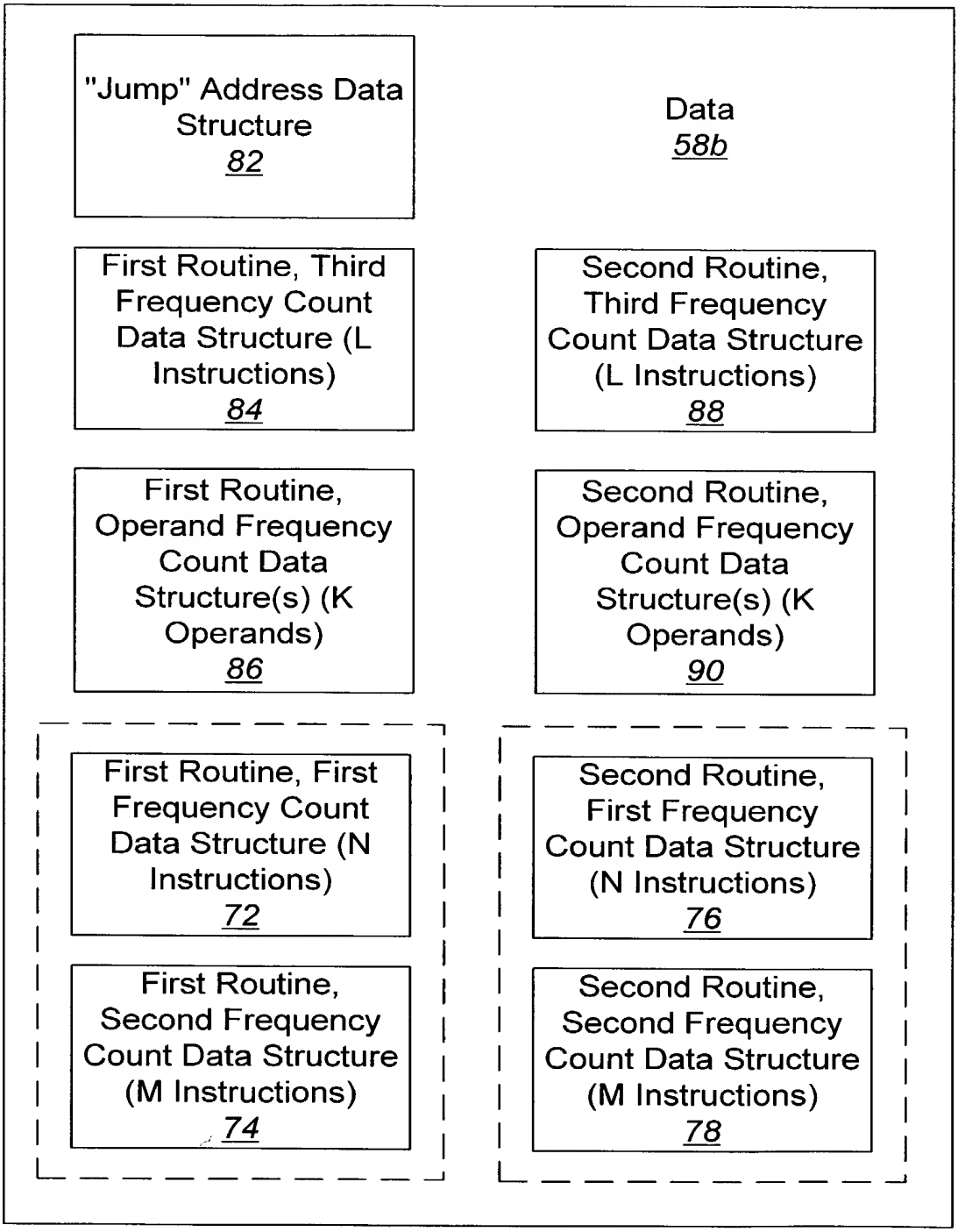

"Jump" Address Data Structure
82
Data
58b
First Routine, Third Frequency Count Data Structure (L Instructions)
84
Second Routine, Third Frequency Count Data Structure (L Instructions)
88
First Routine, Operand Frequency Count Data Structure(s) (K Operands)
86
Second Routine, Operand Frequency Count Data Structure(s) (K Operands)
90
First Routine, First Frequency Count Data Structure (N Instructions)
72
First Routine, Second Frequency Count Data Structure (M Instructions)
74
Second Routine, First Frequency Count Data Structure (N Instructions)
76
Second Routine, Second Frequency Count Data Structure (M Instructions)
78

FIG. 5

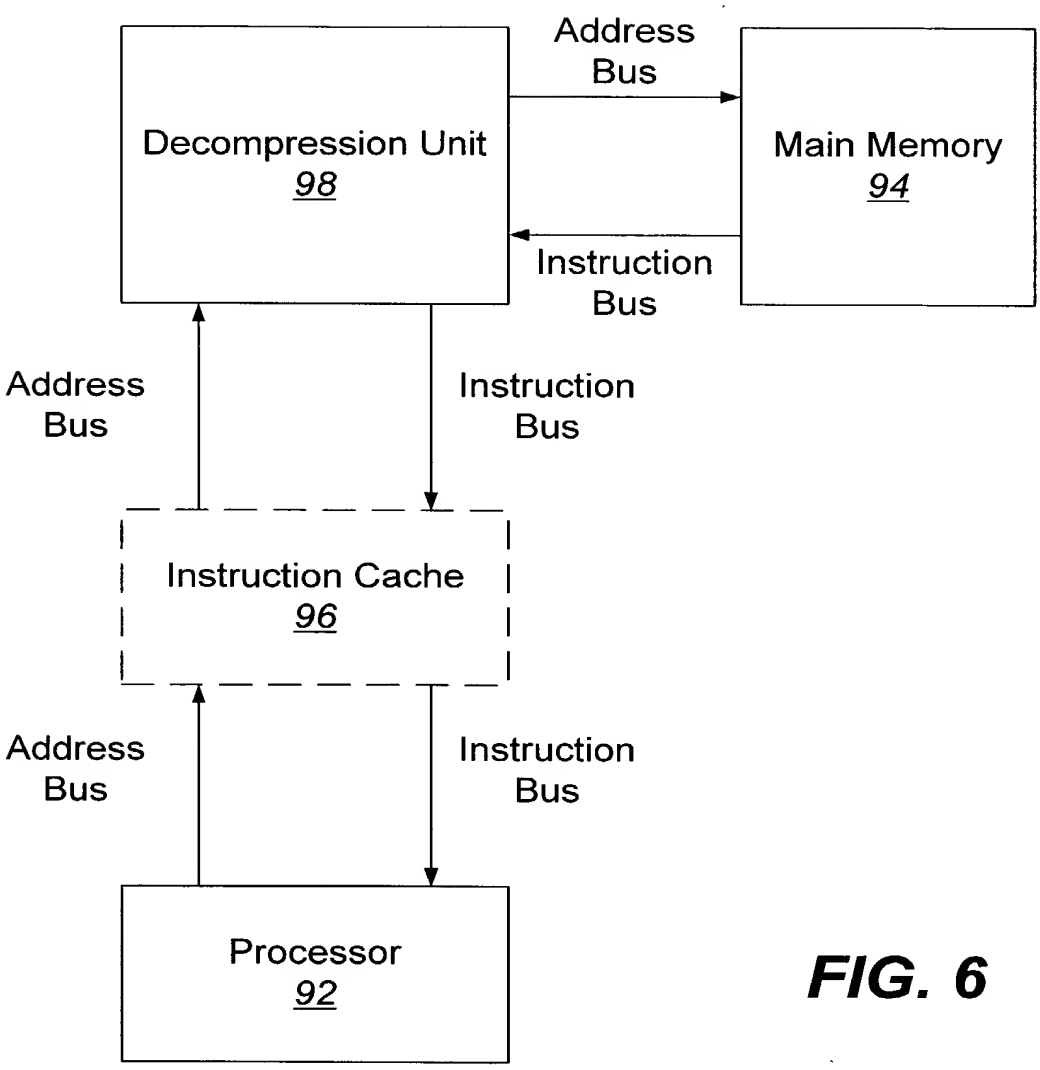

Address
Bus
Decompression Unit
98
Main Memory
94
Instruction
Bus
Address
Bus
Instruction
Bus
Instruction Cache
96
Address
Bus
Instruction
Bus
Processor
92

FIG. 6

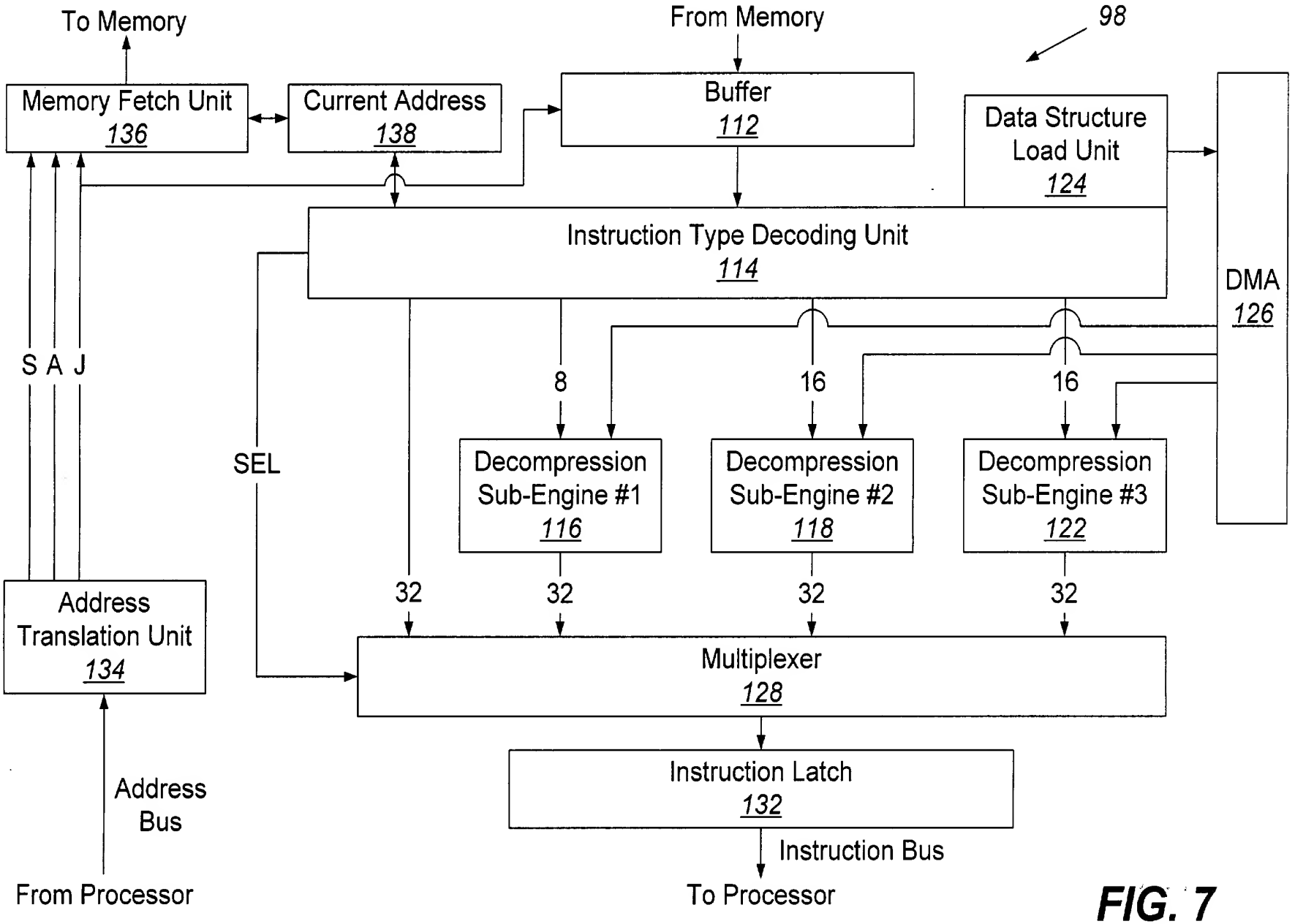

To Memory
From Memory
98
Memory Fetch Unit
136
Current Address
138
Buffer
112
Data Structure
Load Unit
124
Instruction Type Decoding Unit
114
DMA
126
S A J
8
16
16
SEL
Decompression
Sub-Engine #1
116
Decompression
Sub-Engine #2
118
Decompression
Sub-Engine #3
122
32
32
32
32
Address
Translation Unit
134
Multiplexer
128
Instruction Latch
132
Address
Bus
Instruction Bus
From Processor
To Processor

FIG. 7

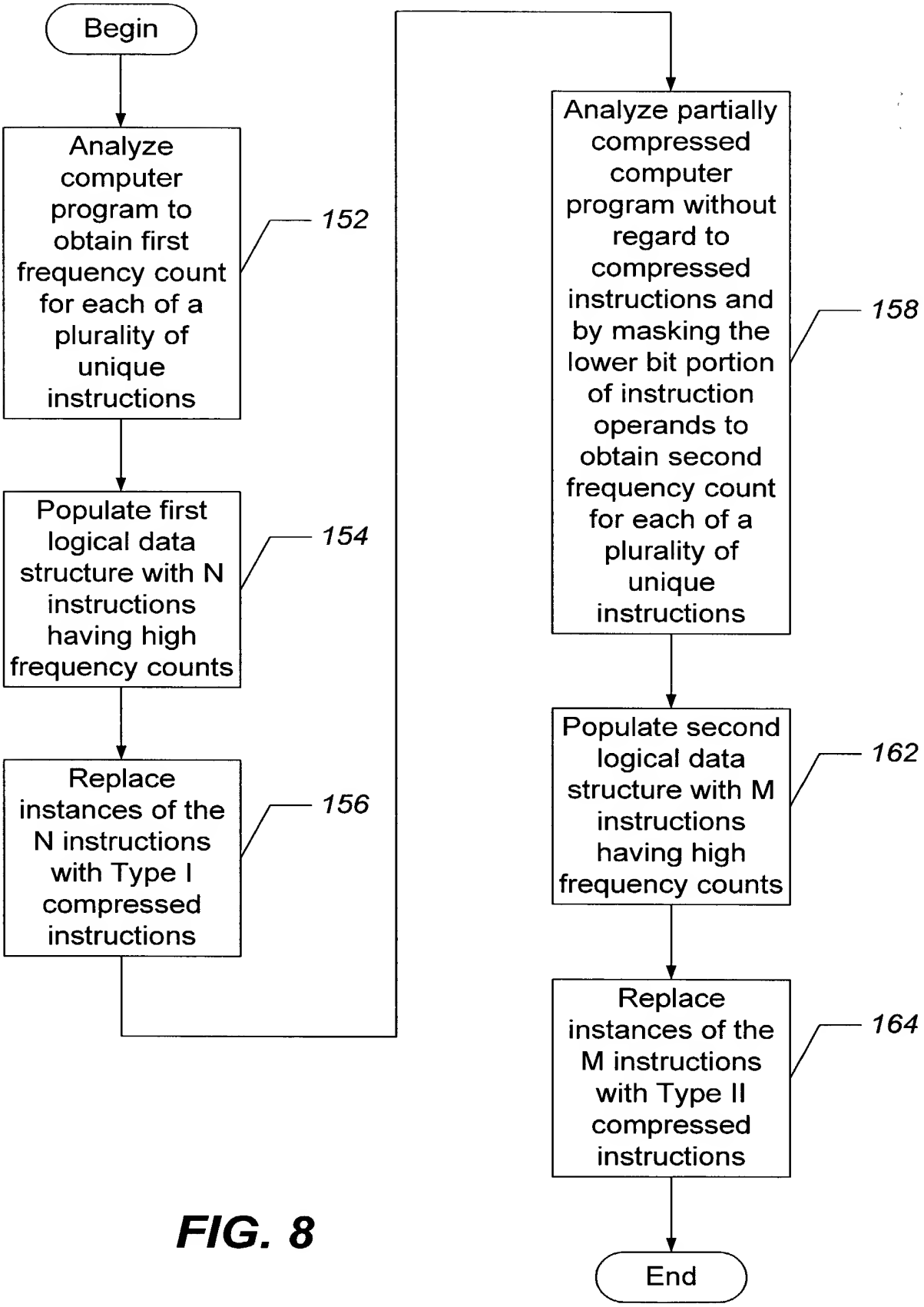

Begin
Analyze computer program to obtain first frequency count for each of a plurality of unique instructions
152
Populate first logical data structure with N instructions having high frequency counts
154
Replace instances of the N instructions with Type I compressed instructions
156
Analyze partially compressed computer program without regard to compressed instructions and by masking the lower bit portion of instruction operands to obtain second frequency count for each of a plurality of unique instructions
158
Populate second logical data structure with M instructions having high frequency counts
162
Replace instances of the M instructions with Type II compressed instructions
164
End

FIG. 8

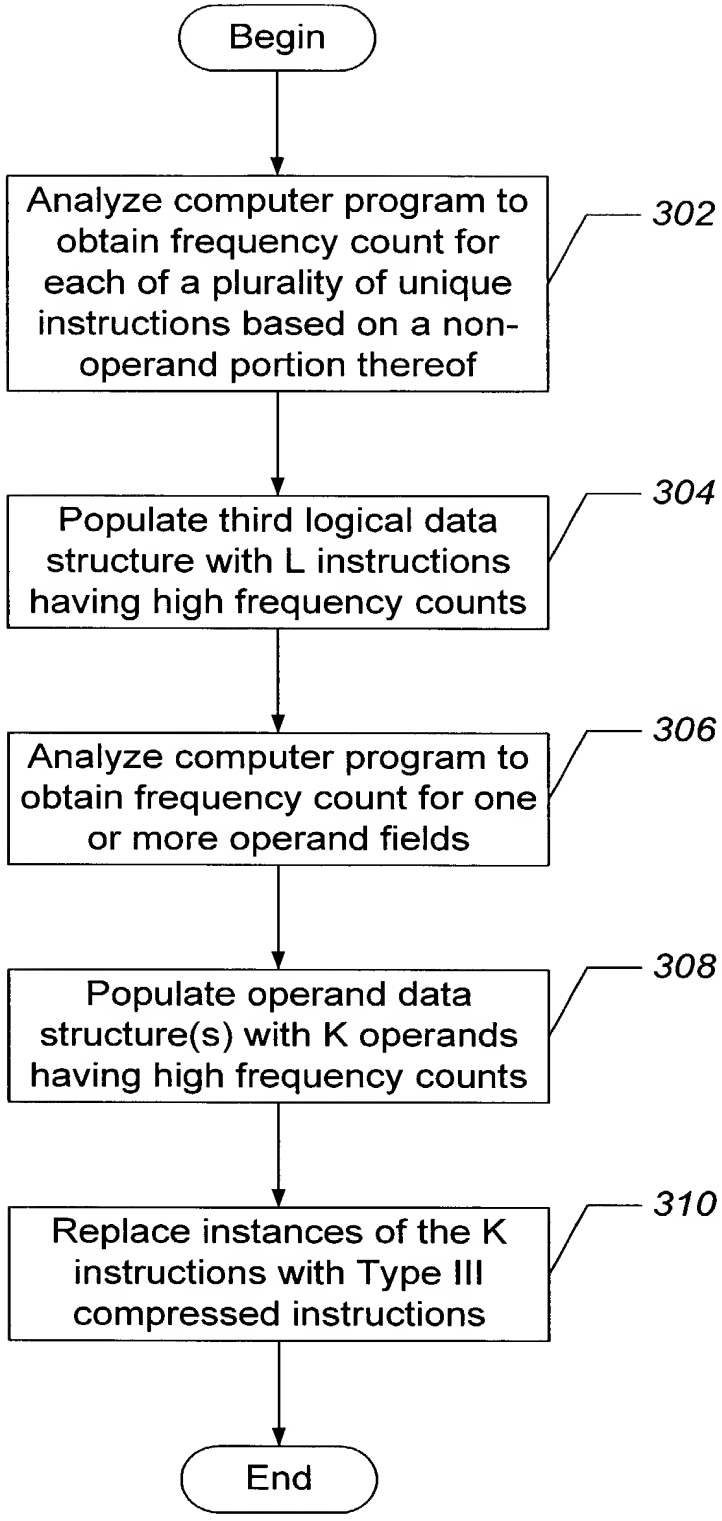

Begin
Analyze computer program to obtain frequency count for each of a plurality of unique instructions based on a non-operand portion thereof
302
Populate third logical data structure with L instructions having high frequency counts
304
Analyze computer program to obtain frequency count for one or more operand fields
306
Populate operand data structure(s) with K operands having high frequency counts
308
Replace instances of the K instructions with Type III compressed instructions
310
End

FIG. 9

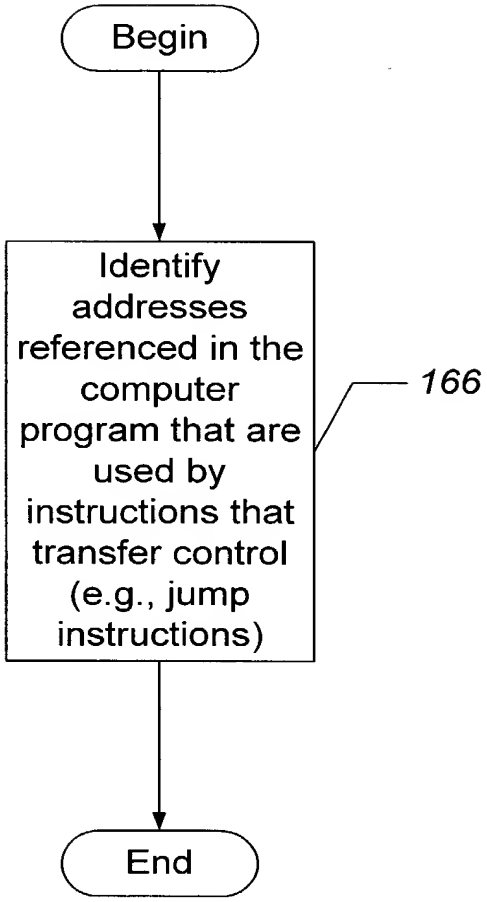

Begin
Identify addresses referenced in the computer program that are used by instructions that transfer control (e.g., jump instructions)
166
End

FIG. 10

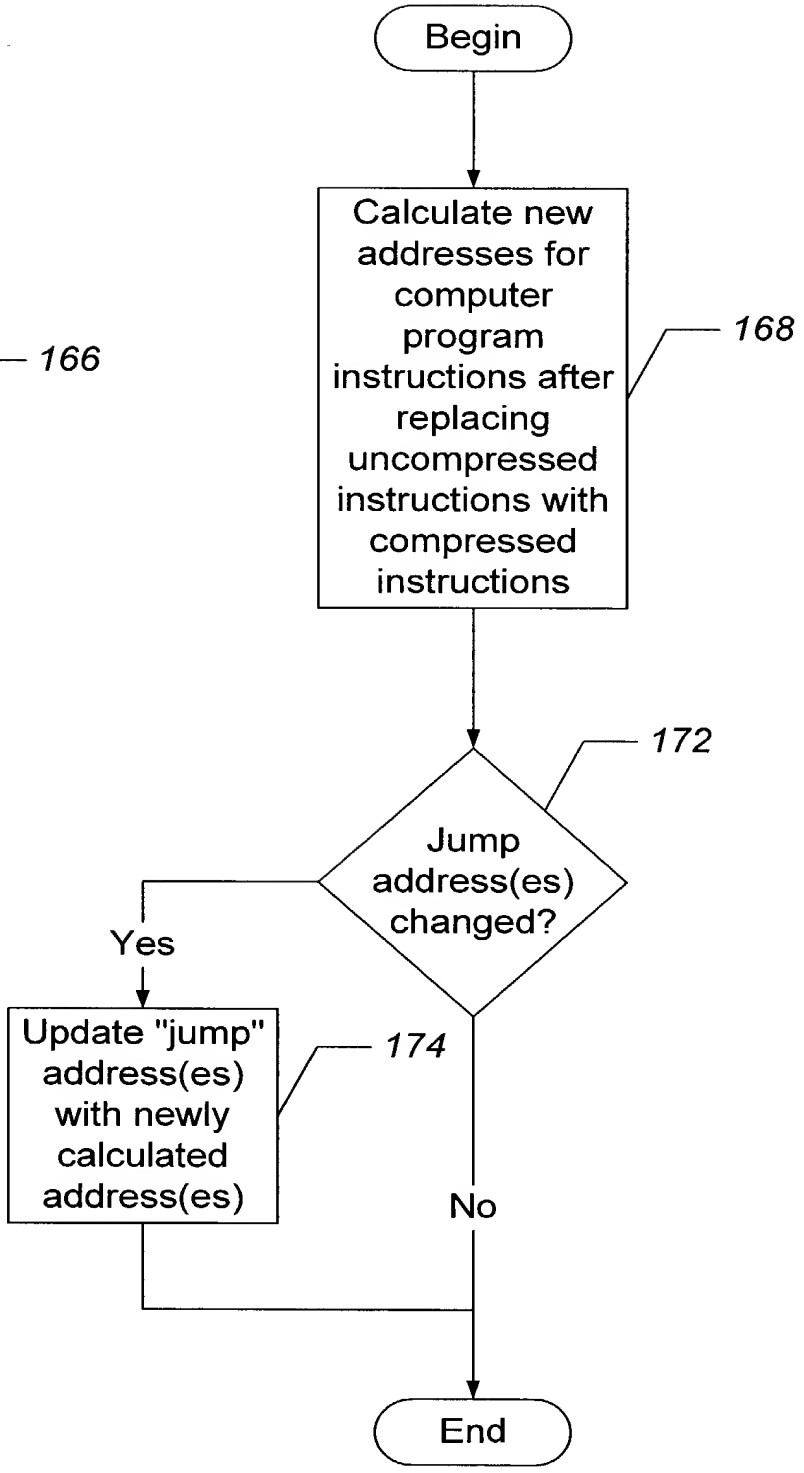

Begin
Calculate new addresses for computer program instructions after replacing uncompressed instructions with compressed instructions
168
Jump address(es) changed?
172
Yes
Update "jump" address(es) with newly calculated address(es)
174
No
End

FIG. 11

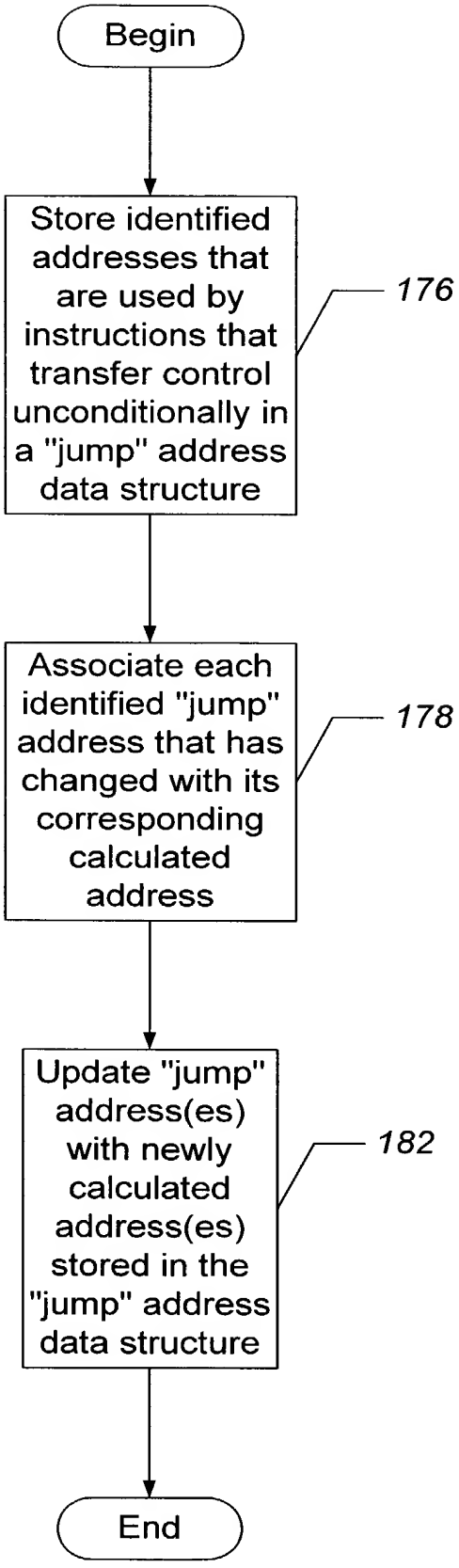

Begin
Store identified addresses that are used by instructions that transfer control unconditionally in a "jump" address data structure
176
Associate each identified "jump" address that has changed with its corresponding calculated address
178
Update "jump" address(es) with newly calculated address(es) stored in the "jump" address data structure
182
End

FIG. 12

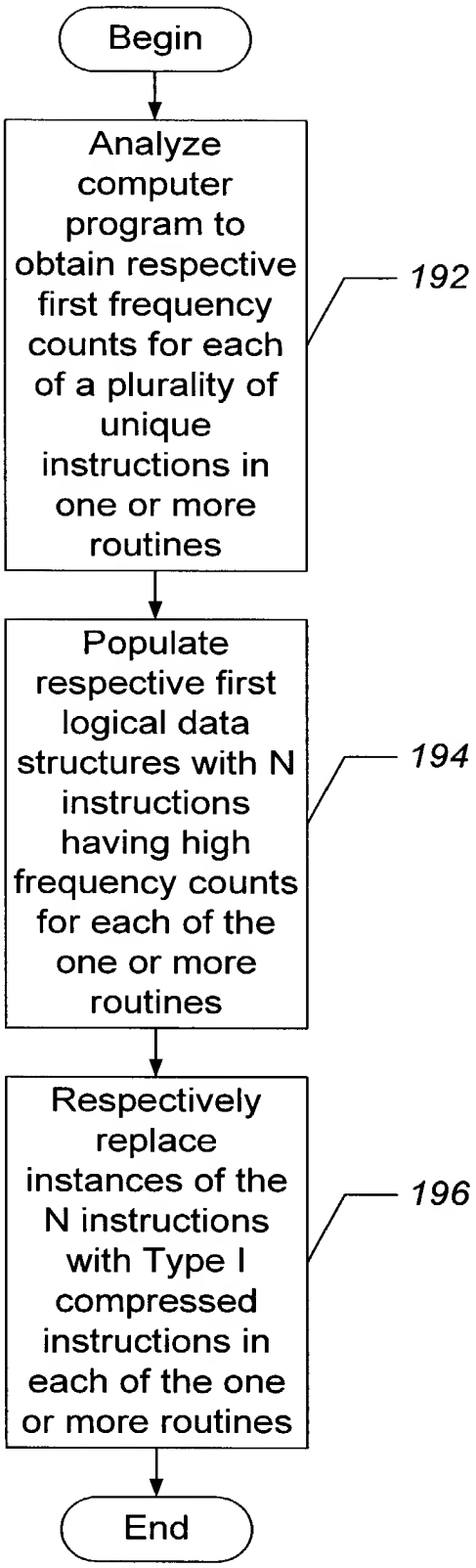

Begin
Analyze computer program to obtain respective first frequency counts for each of a plurality of unique instructions in one or more routines
192
Populate respective first logical data structures with N instructions having high frequency counts for each of the one or more routines
194
Respectively replace instances of the N instructions with Type I compressed instructions in each of the one or more routines
196
End

FIG. 13

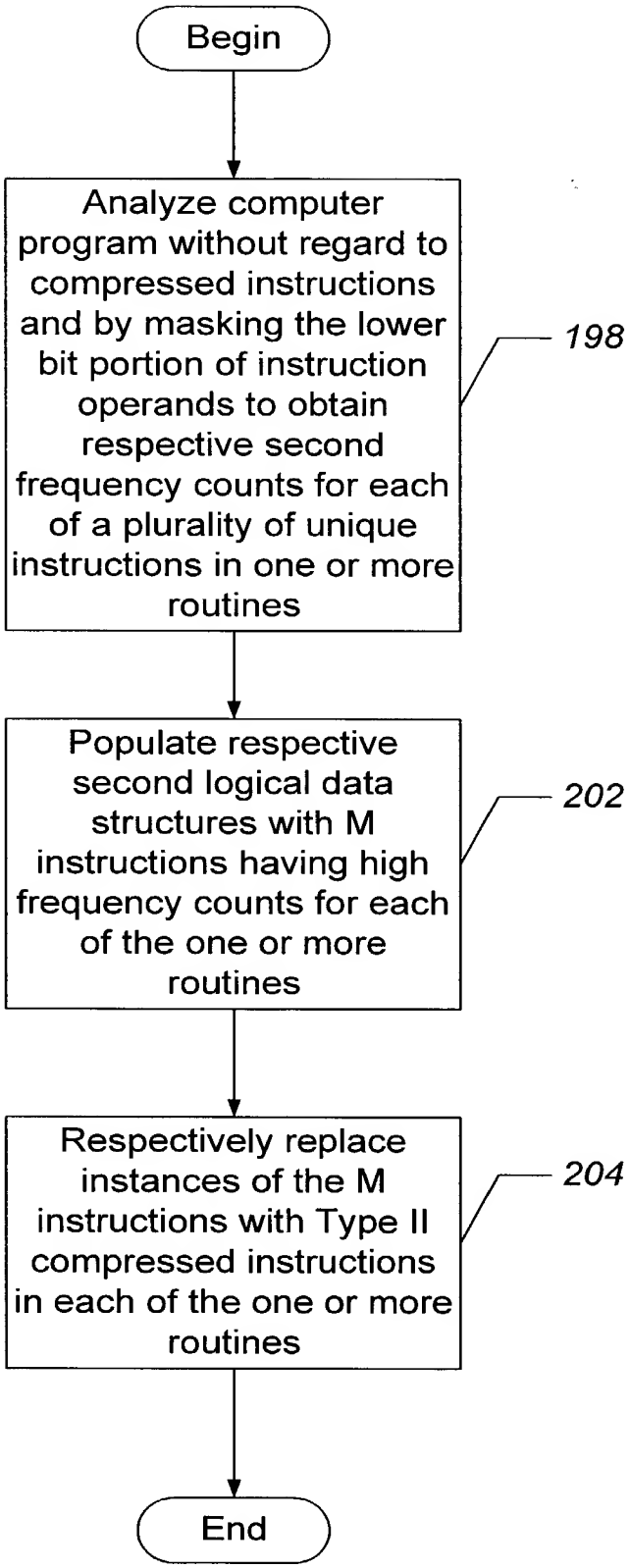

Begin
Analyze computer program without regard to compressed instructions and by masking the lower bit portion of instruction operands to obtain respective second frequency counts for each of a plurality of unique instructions in one or more routines
198
Populate respective second logical data structures with M instructions having high frequency counts for each of the one or more routines
202
Respectively replace instances of the M instructions with Type II compressed instructions in each of the one or more routines
204
End

FIG. 14

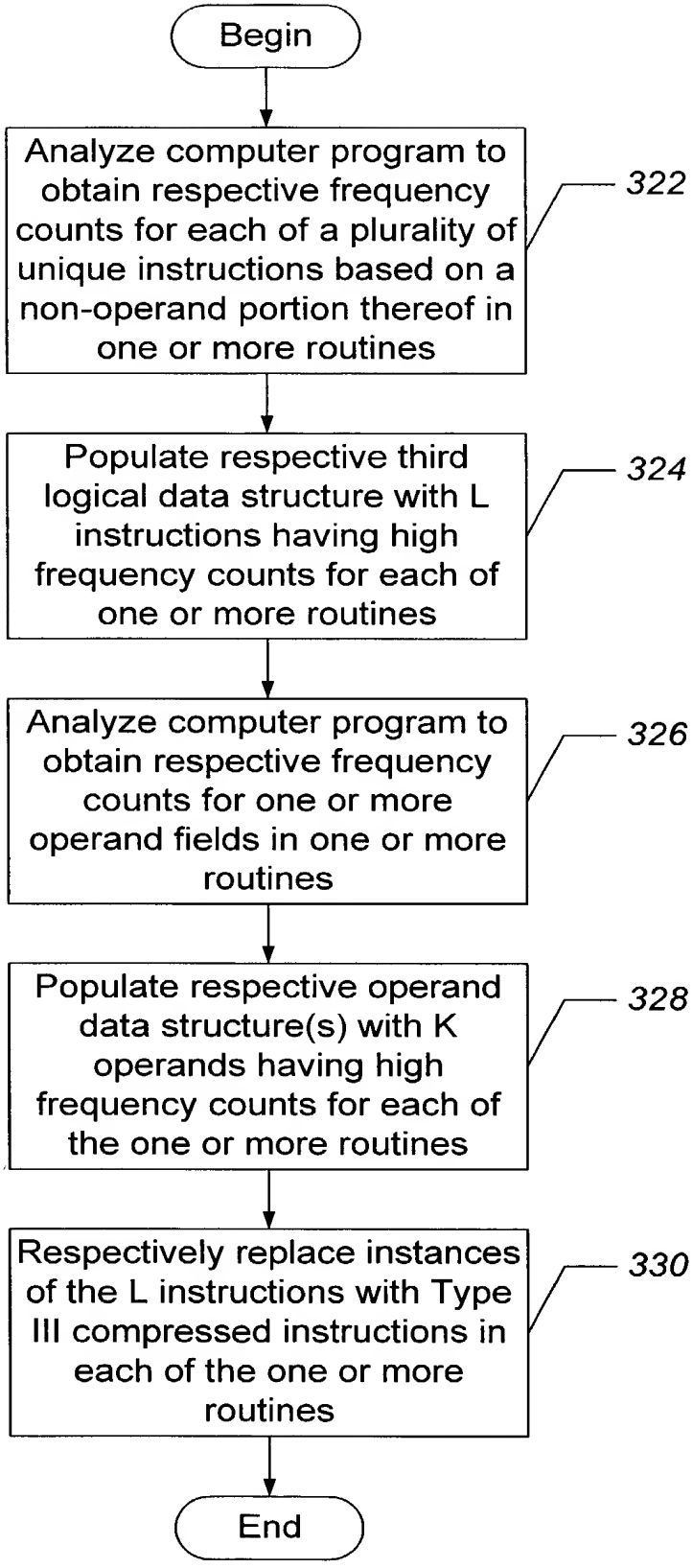

Begin
Analyze computer program to obtain respective frequency counts for each of a plurality of unique instructions based on a non-operand portion thereof in one or more routines
322
Populate respective third logical data structure with L instructions having high frequency counts for each of one or more routines
324
Analyze computer program to obtain respective frequency counts for one or more operand fields in one or more routines
326
Populate respective operand data structure(s) with K operands having high frequency counts for each of the one or more routines
328
Respectively replace instances of the L instructions with Type III compressed instructions in each of the one or more routines
330
End

FIG. 15

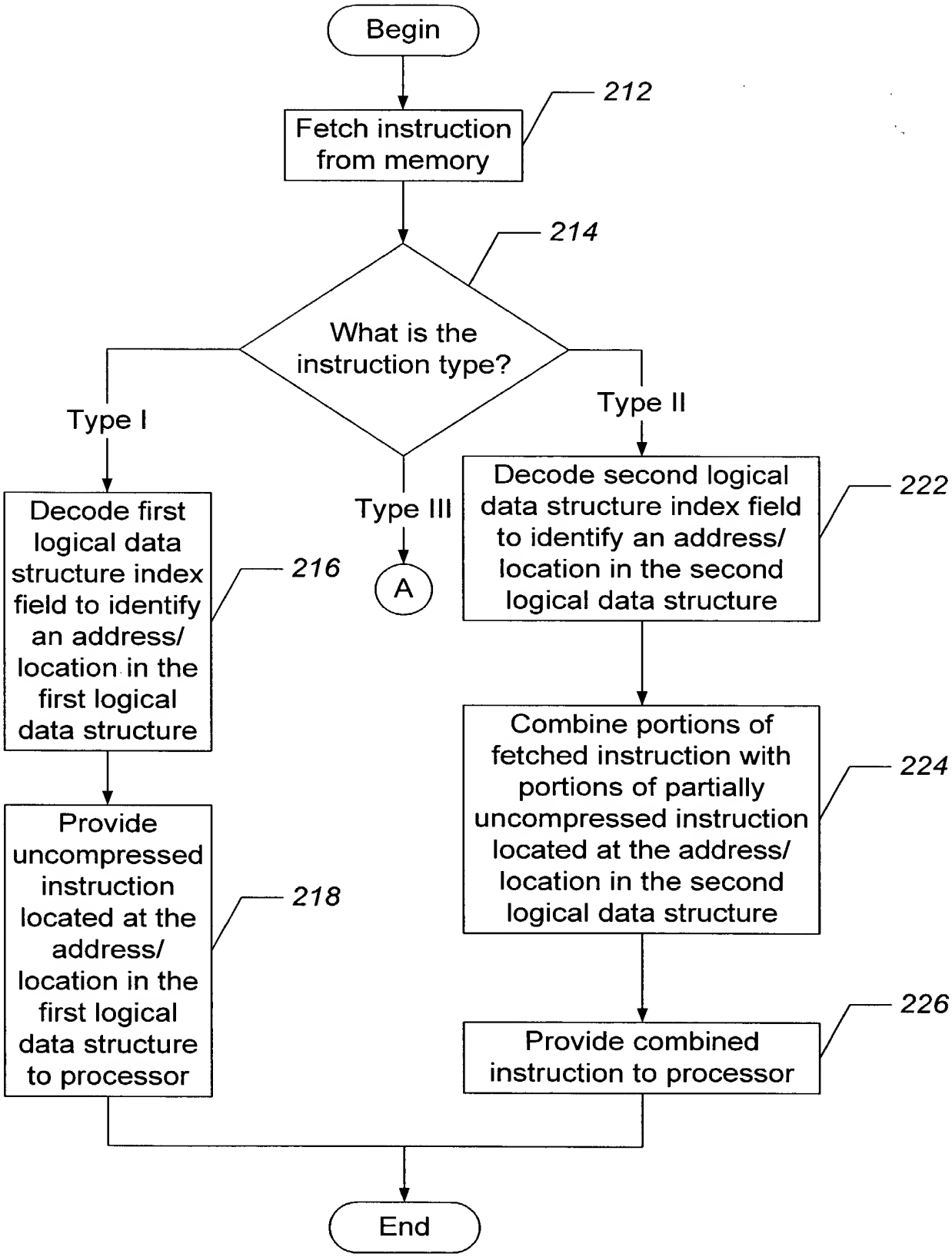

Begin
Fetch instruction from memory
212
What is the instruction type?
214
Type I
Type II
Type III
A
Decode first logical data structure index field to identify an address/ location in the first logical data structure
216
Provide uncompressed instruction located at the address/ location in the first logical data structure to processor
218
Decode second logical data structure index field to identify an address/ location in the second logical data structure
222
Combine portions of fetched instruction with portions of partially uncompressed instruction located at the address/ location in the second logical data structure
224
Provide combined instruction to processor
226
End

FIG. 16A

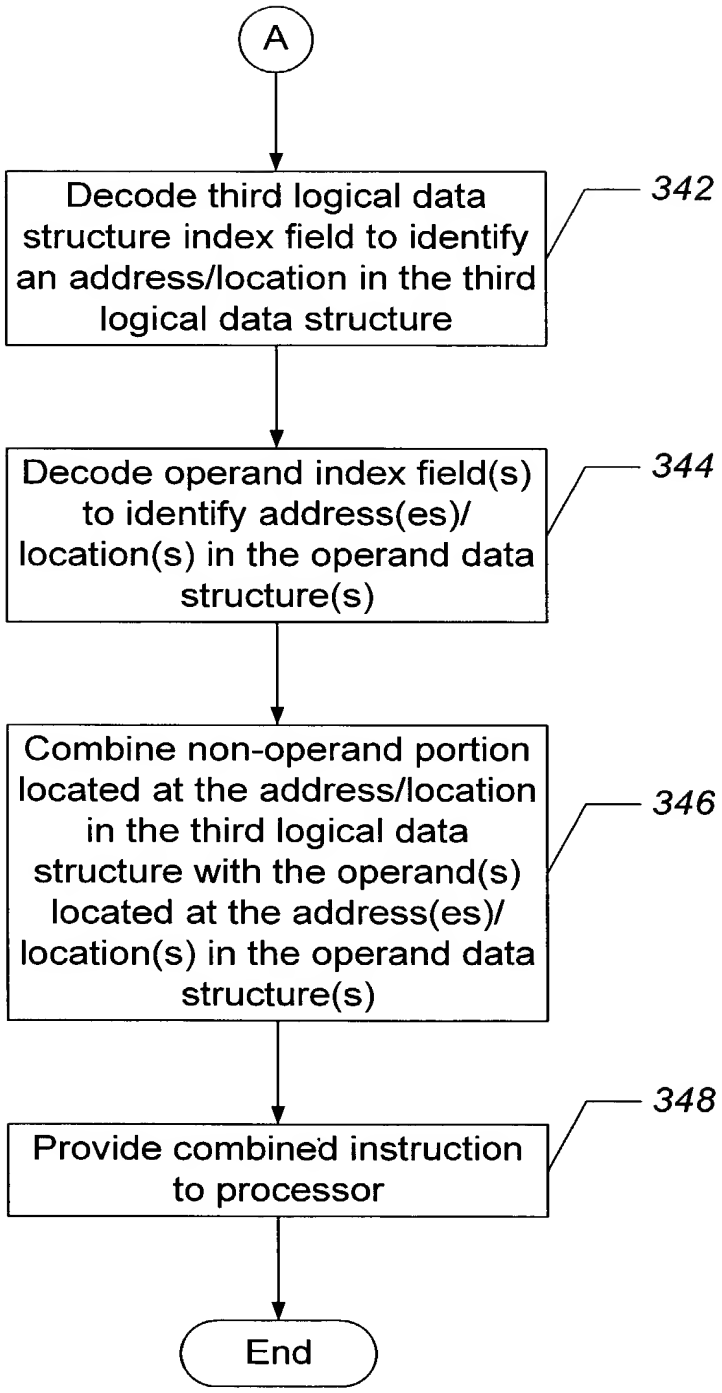

A
Decode third logical data structure index field to identify an address/location in the third logical data structure
342
Decode operand index field(s) to identify address(es)/ location(s) in the operand data structure(s)
344
Combine non-operand portion located at the address/location in the third logical data structure with the operand(s) located at the address(es)/ location(s) in the operand data structure(s)
346
Provide combined instruction to processor
348
End

FIG. 16B

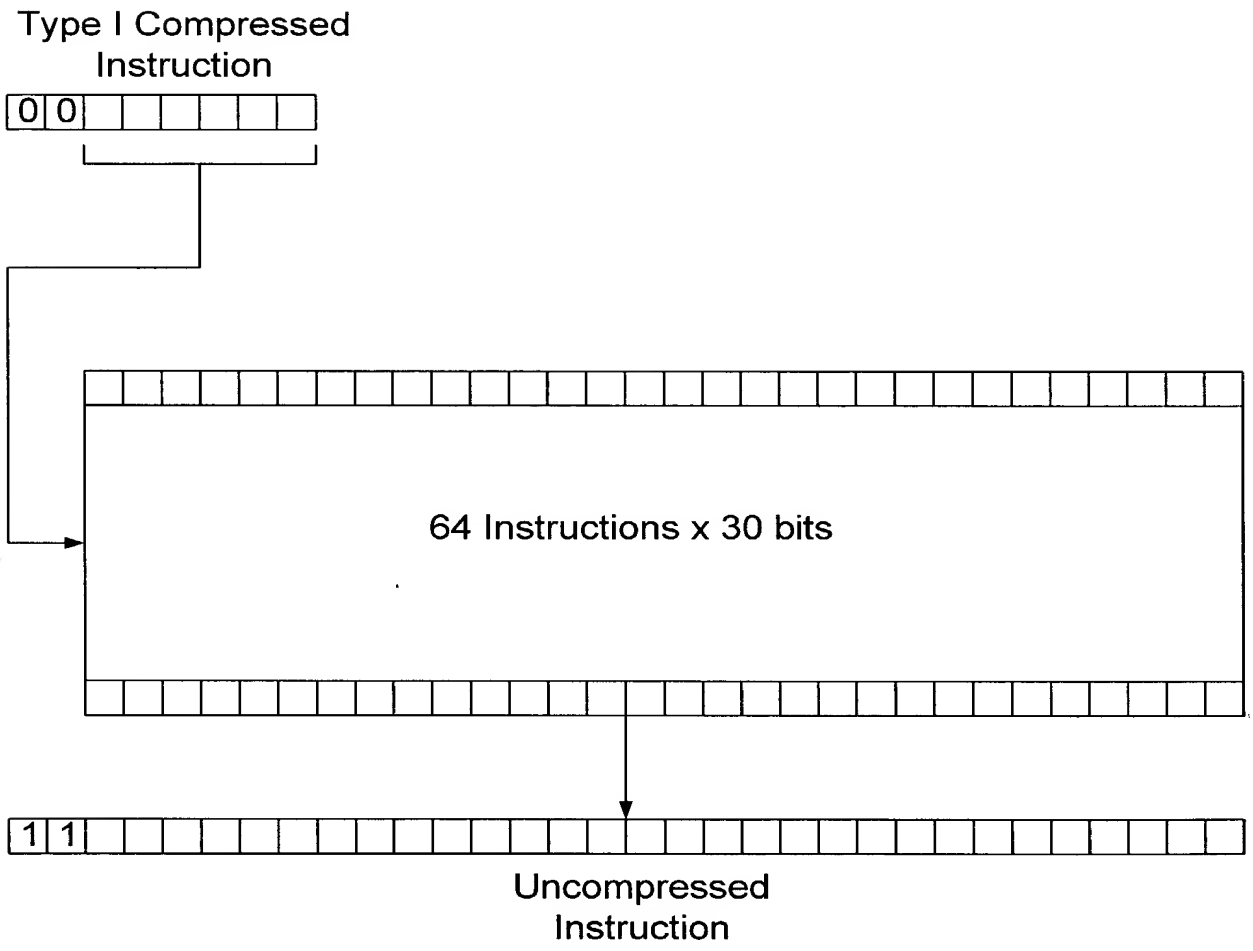

Type I Compressed Instruction
0 0
64 Instructions x 30 bits
1 1
Uncompressed Instruction

FIG. 17

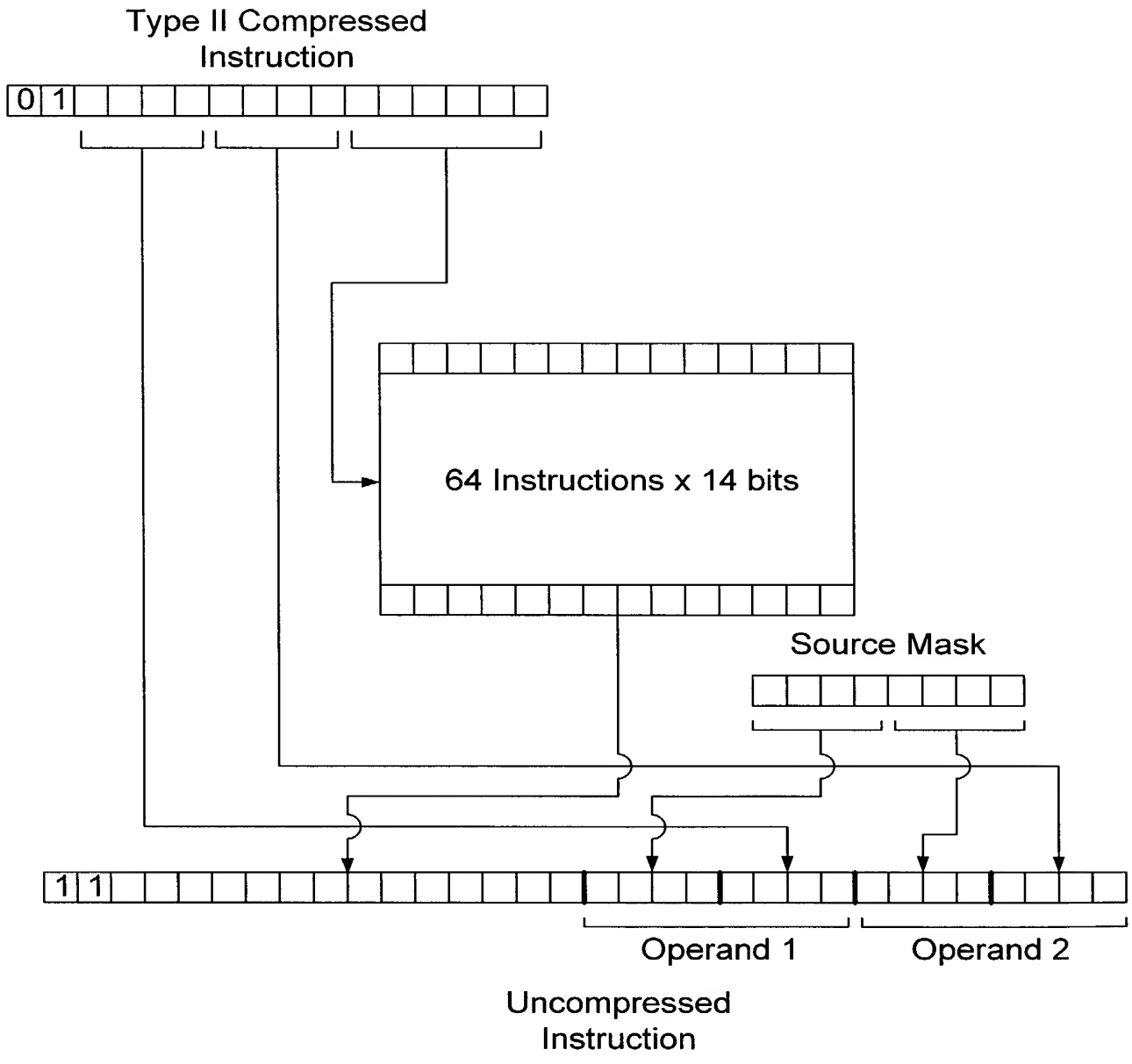

Type II Compressed Instruction
0 1
64 Instructions x 14 bits
Source Mask
1 1
Operand 1
Operand 2
Uncompressed Instruction

FIG. 18

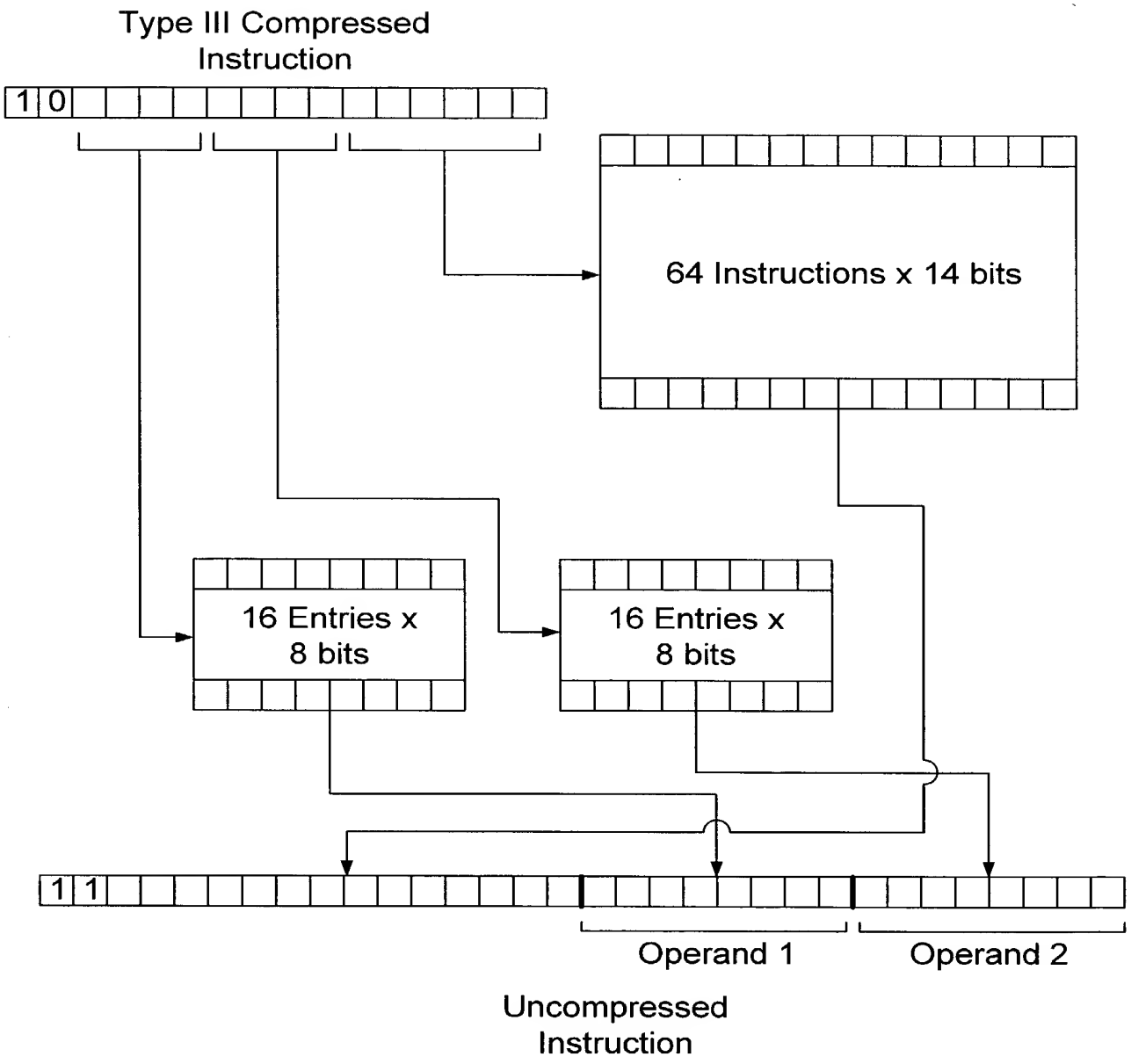

Type III Compressed Instruction
1 0
64 Instructions x 14 bits
16 Entries x 8 bits
16 Entries x 8 bits
1 1
Operand 1
Operand 2
Uncompressed Instruction

FIG. 19

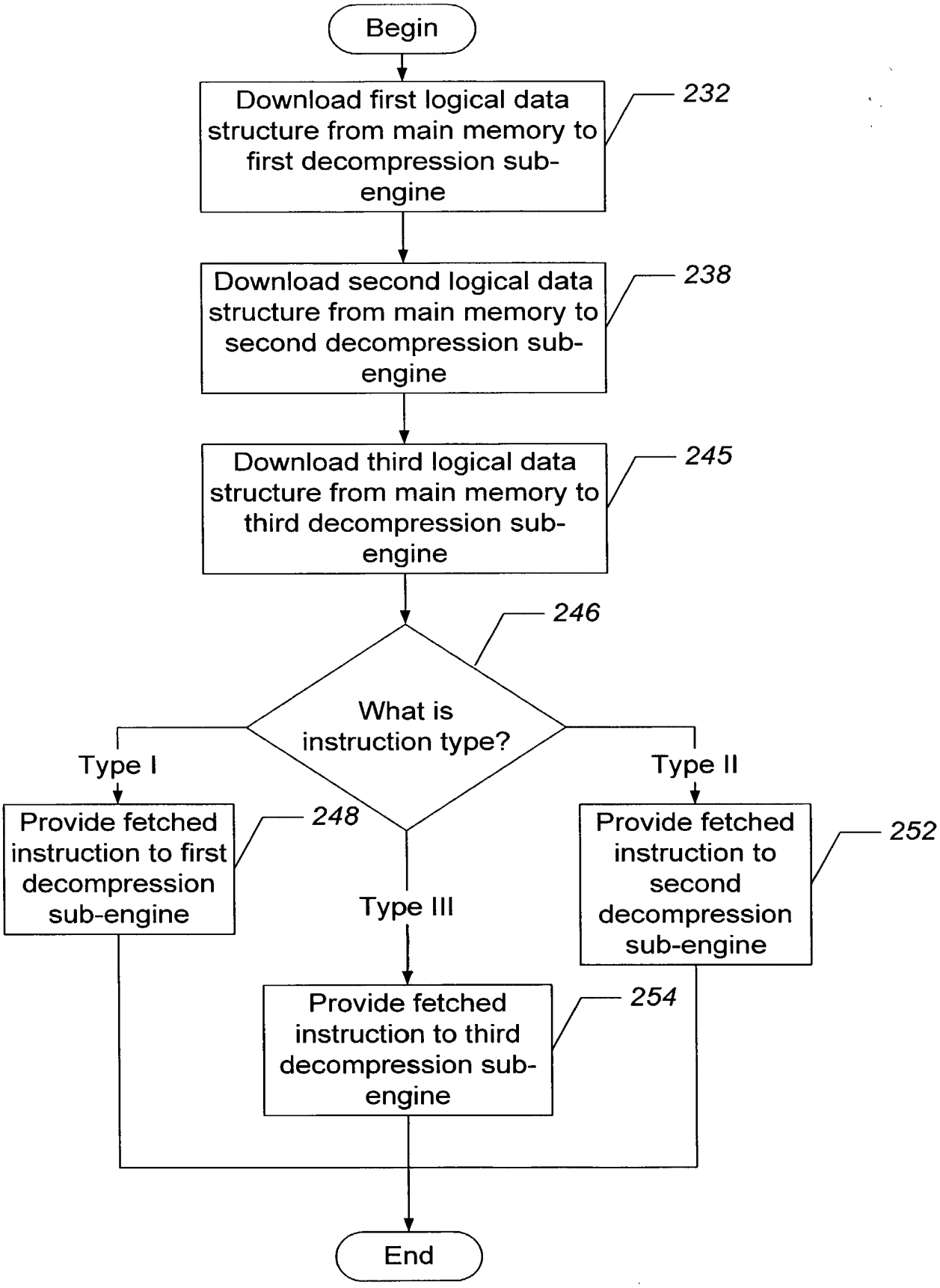

Begin
Download first logical data structure from main memory to first decompression sub-engine
232
Download second logical data structure from main memory to second decompression sub-engine
238
Download third logical data structure from main memory to third decompression sub-engine
245
246
What is instruction type?
Type I
Type II
Type III
Provide fetched instruction to first decompression sub-engine
248
Provide fetched instruction to second decompression sub-engine
252
Provide fetched instruction to third decompression sub-engine
254
End

FIG. 20

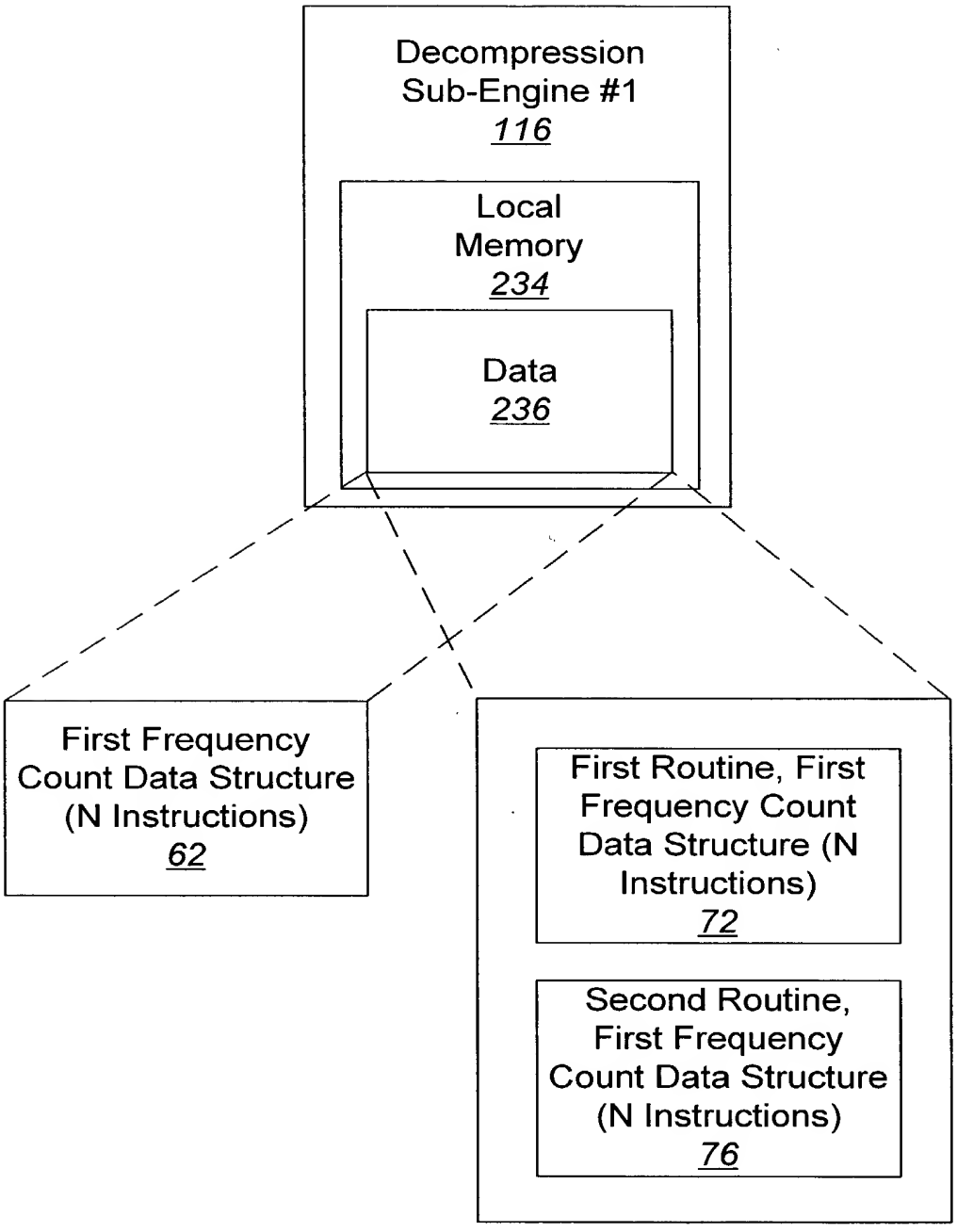

Decompression
Sub-Engine #1
116
Local
Memory
234
Data
236
First Frequency
Count Data Structure
(N Instructions)
62
First Routine, First
Frequency Count
Data Structure (N
Instructions)
72
Second Routine,
First Frequency
Count Data Structure
(N Instructions)
76

FIG. 21

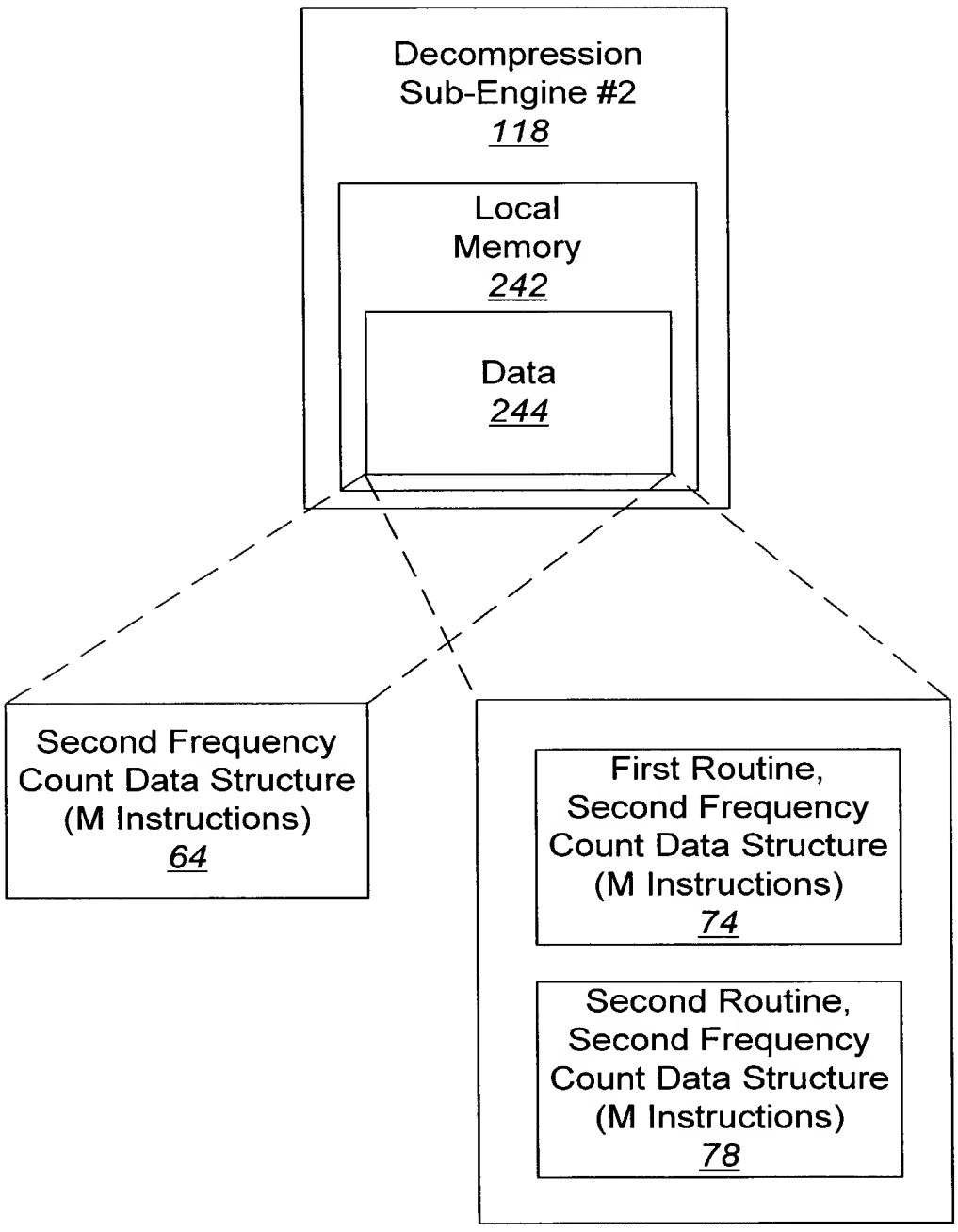

Decompression
Sub-Engine #2
118
Local
Memory
242
Data
244
Second Frequency
Count Data Structure
(M Instructions)
64
First Routine,
Second Frequency
Count Data Structure
(M Instructions)
74
Second Routine,
Second Frequency
Count Data Structure
(M Instructions)
78

FIG. 22

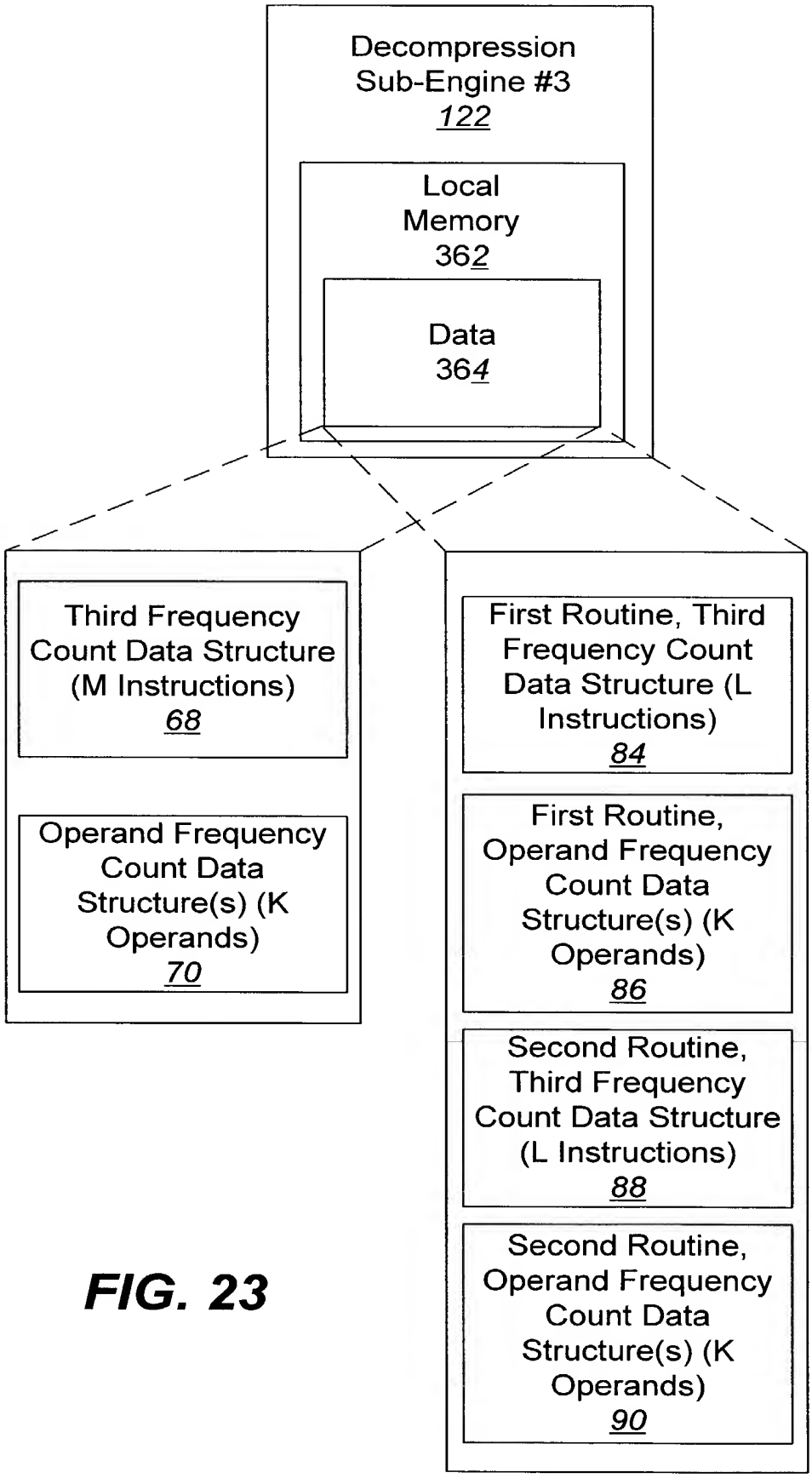

Decompression Sub-Engine #3 122
Local Memory 362
Data 364
Third Frequency Count Data Structure (M Instructions) 68
Operand Frequency Count Data Structure(s) (K Operands) 70
First Routine, Third Frequency Count Data Structure (L Instructions) 84
First Routine, Operand Frequency Count Data Structure(s) (K Operands) 86
Second Routine, Third Frequency Count Data Structure (L Instructions) 88
Second Routine, Operand Frequency Count Data Structure(s) (K Operands) 90

FIG. 23

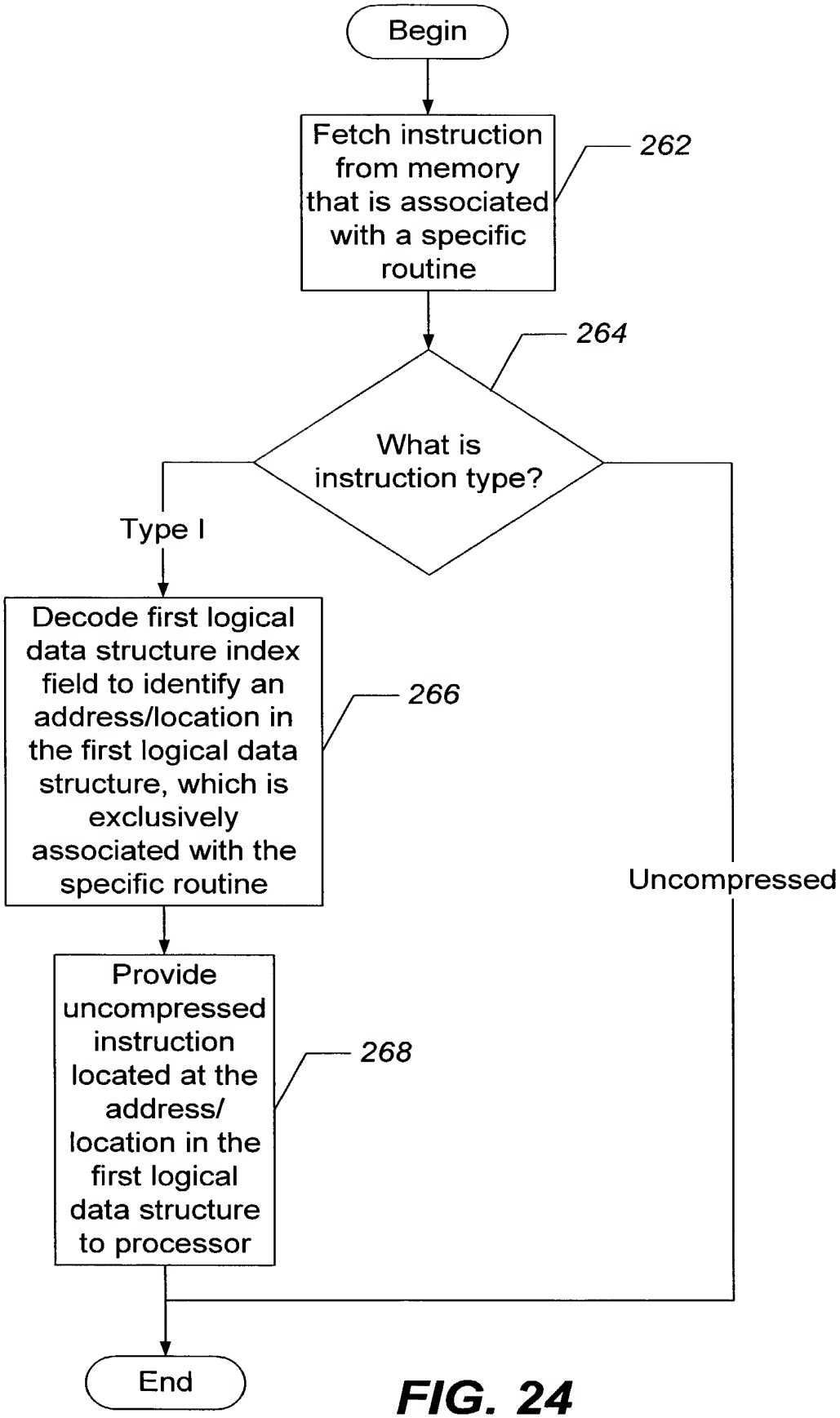

Begin
Fetch instruction from memory that is associated with a specific routine
262
264
What is instruction type?
Type I
Uncompressed
Decode first logical data structure index field to identify an address/location in the first logical data structure, which is exclusively associated with the specific routine
266
Provide uncompressed instruction located at the address/ location in the first logical data structure to processor
268
End

FIG. 24

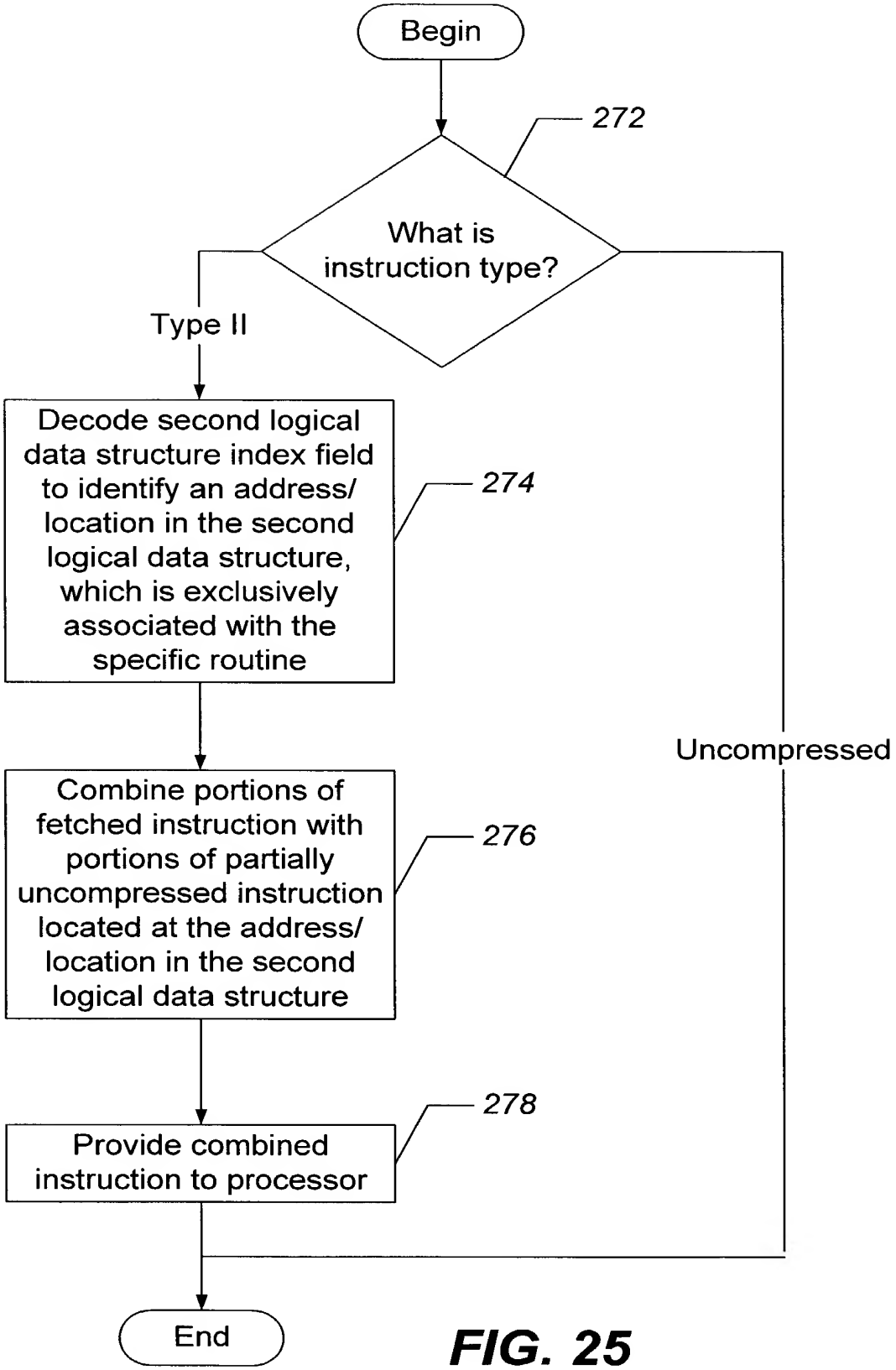

Begin
272
What is instruction type?
Type II
Uncompressed
Decode second logical data structure index field to identify an address/ location in the second logical data structure, which is exclusively associated with the specific routine
274
Combine portions of fetched instruction with portions of partially uncompressed instruction located at the address/ location in the second logical data structure
276
Provide combined instruction to processor
278
End

FIG. 25

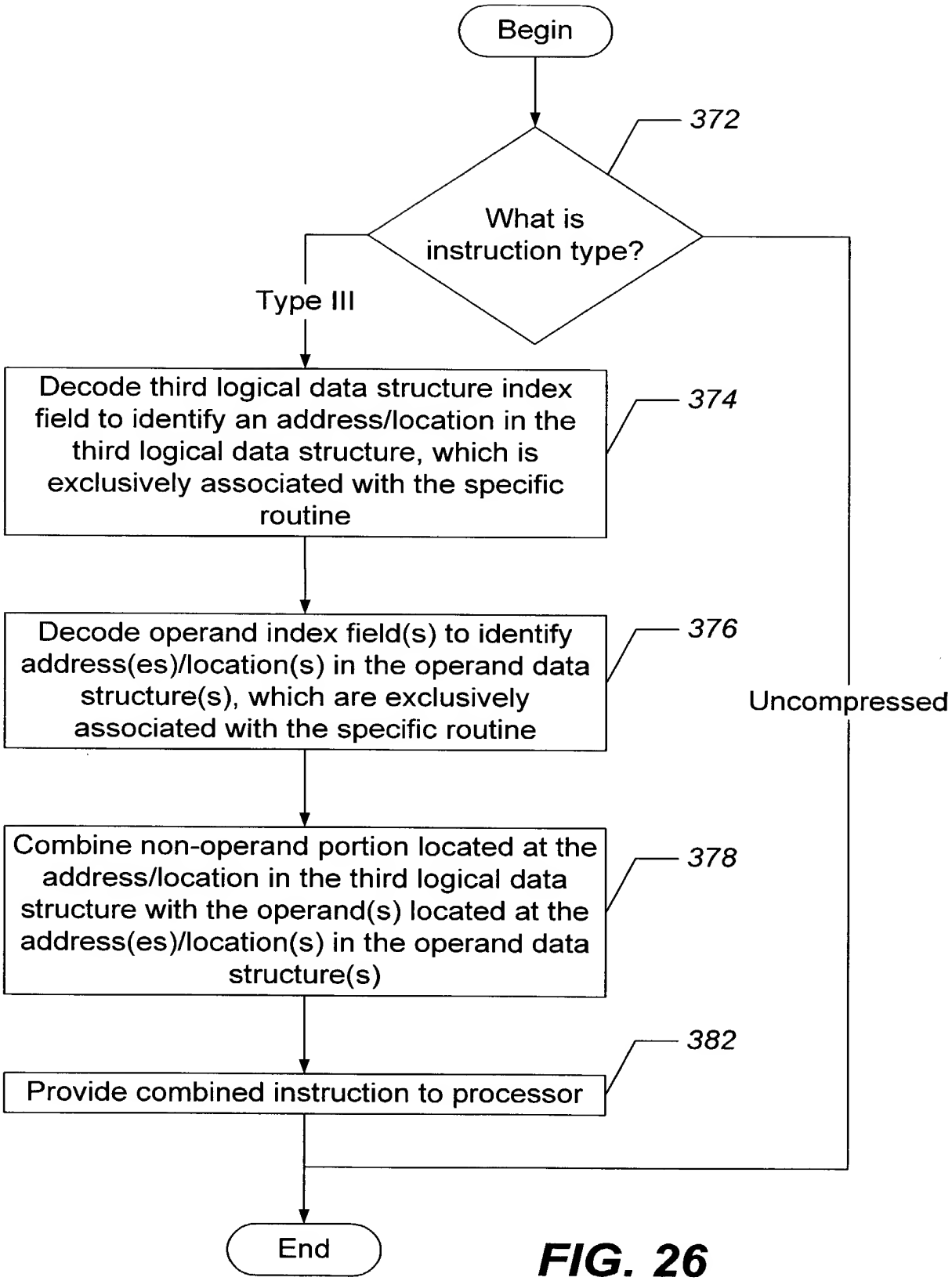

Begin
372
What is instruction type?
Type III
Uncompressed
Decode third logical data structure index field to identify an address/location in the third logical data structure, which is exclusively associated with the specific routine
374
Decode operand index field(s) to identify address(es)/location(s) in the operand data structure(s), which are exclusively associated with the specific routine
376
Combine non-operand portion located at the address/location in the third logical data structure with the operand(s) located at the address(es)/location(s) in the operand data structure(s)
378
382
Provide combined instruction to processor
End

FIG. 26